张量数据中的多密集块检测方法 *

范卫俊，程艳云

(南京邮电大学 自动化学院，南京 310023)

摘　要： 过去的许多研究表明在实际张量数据中密集的部分存在着异常或者欺诈行为，如微博僵尸粉行为、网络攻击等。因此，研究人员提出了各种各样的方法来针对密集块的提取，但是这些方法存在着低准确率和召回率的缺点。针对这些缺点，提出了一种基于二叉树搜索的多密集块检测方法，简称 DDB-BST，通过对张量数据进行基于评价指标的局部搜索，找到评价指标最高的子张量数据，将数据分成左右子节点，通过不断比较父节点和左右子节点评价指标值的数值关系，判断二叉树生长是否终止。对终止条件给出了严格的数学证明。在合成数据集以及真实数据集上进行实验，发现 DDB-BST 比现有的 M-zoom 多密集块方法的 F1 值提高近 30%。

关键词： 张量数据；密集块 ；二叉树搜索；终止条件

中图分类号： TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.08.0866

Method on dense-blocks detection in tensor data

Fan Weijun, Cheng Yanyun

(*College of Automation*, *Nanjing University of Posts & Telecommunications*, *Nanjing* 310023, *China*)

Abstract: Past studies have shown that dense blocks in real-world tensor can have anomalous or fraudulent behavior such as zombie followers' behavior or network attack. Thus, various methods have been used for detecting dense blocks in tensor. However, these methods have low accuracy or low recall rate. To overcome those limitations, DDB-BST is proposed, a method on dense blocks detection based on binary tree search which finds the block with the highest metric in tensor by local search. By comparing key values between child's and father's nodes, we judge whether the binary tree grows. Finally, when the binary tree stop growing, all the child nodes are dense blocks. End condition of binary tree growing is mathematically proven. Experiments on both synthesis data and real-world data show efficiency of the method, the F1 value with DDB-BST being 30 percent higher than for M-zoom.

Key Words: tensor data; dense-blocks; binary tree search; end condition

0　引言

假设你的工作是检测异常的网络连接，那么如何提取出如下的异常行为：一群 IP 地址每几秒向同一个 IP 地址中的某几个端口发送连接请求或者是得到一段时间内网络连接的特征信息如协议类型、从源到目的地的字节数、过去两秒内与当前连接相同主机的连接数等。目前很多研究已经通过张量模型[1]来处理这类问题，而在张量中的密集块代表着一群用户的同步行为，而这些行为往往是可疑的。所以张量中的密集块提取被广泛的应用于网络入侵检测[2]、提高微博转发量检测[3]、僵尸粉活动[4]以及遗传学[1]。现在主要有三种类型的方法能对张量中的密集块进行快速准确的检测。一种是基于张量分解的密集块挖掘，张量分解应用于对密集子张量的挖掘，如 HOSVD[1]和 CP 分解。近年来研究者们对张量分解的方法不断改进，如基于分布模型的方法[5]；基于采样的方法[6]以及针对数亿规模数据的张量分解[7]等。然而基于张量分解的密集块挖掘方法存在着几点缺点：没考虑背景数据的性质；在密度指标下不具有较高延展性；不能提供合理的边界。另一种是稠密子图挖掘方法，文献[8]对稠密子图的挖掘方法进行了总结。最新的稠密子图挖掘的方法主要有：采用最大整体密度和有限重合寻找密集子图[9]；基于核分解的稠密子图的发现[10]；基于一种新的评价指标来发现在不确定图上的稠密子图[11]以及基于数据流或者分布式的动态稠密子图的挖掘[12-13]。最新的研究是通过定义新的标准来对张量中的密集块的检测，文献[14]提出的 CrossSpot 算法是通过随机选取一个块，然后使用一种近似于贪心的方法不断调整这个块的维度，直到其达到局部最优。由于 CrossSpot 没有提供一个合适的边界，所以在面对存在多密集块的张量数据中，检测效果并不好。文献[15]提出的 M-zoom 算法同样使用一种

基金项目： 国家自然科学基金资助项目（61573194）；江苏省自然科学基金青年项目（BK20150851）

作者简介： 范卫俊，硕士研究生，主要研究方向为网络用户异常行为（fanweijun1992@126.com）；程艳云，副教授，主要研究方向为网络优化，行为分析.

近似于贪心的方法，从整个张量数据开始不断移除每个维度中的值直到其达到局部最优即最优部分为一个密集块，然后将密集块移除，再对剩下的进行重复操作。虽然这样可以达到多密集块检测的目的，但是其准确率大大降低。

为了克服上述基于评价指标算法存在的问题，本文提出了一种基于 Suspiciousness 评价指标的二叉树多密集块搜索算法 (Detect Dense Block with Binary Search Tree，DDB-BST)，可以有效的解决在张量数据中对不同形式的多密集块的检测。通过严格的数学证明证明了二叉树的生长条件，从而实现对二叉树中根数据是否存在密集块的判定。应用不同数据集的实验结果表明，本文提出的算法可以有效地对张量数据进行密集块检测。

1 符号及定义

1.1 符号定义

表 1 罗列出文章所出现的符号以及其说明。

表 1 符号说明

符号	定义
D(A1,...,AK,X)	含有 K 个标称属性和一个非负的数值属性的张量数据
K	D 中标称属性的个数即张量数据的维度
Aj	D 中第 j 个标称属性
X	D 中的数值属性数据
B(B1,...,BK,X)	D 中的子张量数据记录
ρ(B,D)	子张量数据 B 在 D 下的评价指标值
SD（或者是SB）	张量数据 D 中 X 的和（或者是子张量数据 B 中 X 的和）
VD（或者是VB）	张量 D 的体积（或者是子张量数据 B 的体积）
aj i	在 Aj 中第 i 个值
Δsaj i	在 Aj 中第 i 个值下所有 X 的和

例 1 网络访问历史记录。在图 1 中，定义了一组数据 *D*(user ip date count)并且每一条数据 d（u, IP, d, c）表示着用户 u 在时刻 d 访问了 IP 地址 c 次，所以数据 *D* 中的标称属性 A_1 = user，A_2 = IP，A_3 = date 和数值属性数据 *X* = count。令 B_1 = {Lucy，Nike}，B_2 = {192.168.5.2,192.168.2.1} 和 B_3 = {2017-1-2}，所以 *B* 表示的子张量是图 1(b)中的阴影部分，此时对应的符号的值 S_B = 16，V_B = 4，$a1\ 1$ = {5}，$\Delta s_{a1\ 1}$ = {7,3,1}。

1.2 评价指标

本文选择文献[16]提出的 Suspiciousness 评价指标。

定义 1（Suspiciousness）：在张量数据 D 中子张量数据 B 的 Suspiciousness 值的计算公式如下：

$$\rho(B,D) = S_B(\ln\frac{S_B}{S_D} - 1) + S_D\frac{V_B}{V_D} - S_B\ln\frac{V_B}{V_D} \quad (1)$$

Suspiciousness 评价指标可以很好的满足公理 1、2。

公理 1 密度公理。如果两个子数据框的尺寸一样，那么这两个子数据框中数值属性的所有值的和越大，则 Suspiciousness 的值越大，即

$$S_B > S_{B'} \text{且} |B_n| = |B'_n|,\ \forall n \in K \Rightarrow \rho(B,D) > \rho(B',D)$$

公理 2 浓度公理。如果两个子数据框中数值属性的所有值的和一样，那么尺寸越小的 Suspiciousness 的值越大，即：

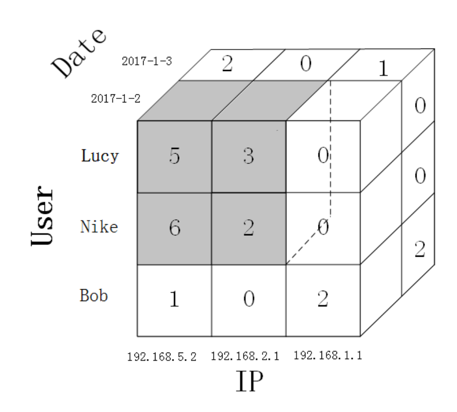
$$V_B < V_{B'} \text{且} S_B = S_{B'} \Rightarrow \rho(B,D) > \rho(B',D)$$

1.3 问题定义

问题 1 给定一个 K 维的张量数据 D，数据中存在这 m 个密集块，采用基于式(1)的评价指标进行二叉树搜索和局域搜索算法找到数据中的 m 个密集块。

User	IP	Date	Count
Lucy	192.168.5.2	2017-1-2	5
Lucy	192.168.2.1	2017-1-2	3
Nike	192.168.5.2	2017-1-2	6
Nike	192.168.1.1	2017-1-2	2
Bob	192.168.1.1	2017-1-3	1
…	…	…	…

(a)一组数据框的数据 ***A***,深色部分的数据表示数据框 ***B***



（b）（a）所对应的张量数据 ***A***，以及其中选中的子张量

图 1 关于例 1 的描述

2 基于评价指标的二叉树搜索的多密集块检测

2.1 基于评价指标的二叉树搜索的多密集块检测算法流程

DDB-DST 算法有两部分组成,首先对原数据进行预处理，得到适用于该算法处理的数据格式；其次对处理后的数据进行算法 2 搜索，得到二叉树的左右子节点；再对左右节点运行算法 2，如此反复直至满足终止条件。这样就可以得到原数据中的多个异常块，具体算法流程所下示：

算法 1 二叉树搜索算法：

输入：原始张量数据 D_{ori}

输出：k 个密集数据块

R = D_{ori},$node_R$ 编号为 0

2.将数据 R 插入到二叉树中，则 *node* 编号为 $node_R$，键值 $key_{根} = \rho(B, D_{ori})$

3.对数据 *R* 进行算法 2 计算，得到二叉树的左节点 B_l 和右

节点 B_r,其 node 编号分别为 $node_R$+“1”和 $node_R$+“0”,键值为

$$key_L=\rho(B_l,D_{ori}) \text{ 和 } key_R=\rho(B_r,D_{ori})$$

4.如果 $key_{根}\le key_L+key_R$，则 $R=B_l$ 继续第二步，$R=B_r$ 继续第二步。

5.如果 $key_{根}>key_L+key_R$，则保存 R 到列表 dense_block 中。

6.全部查找结束，返回 dense_block。

算法 2 局域搜索单密集块算法：

输入：张量数据 R，种子 seed $\tilde{A}=\{\tilde{A}_j\}_{j=1}^{K}$

输出：二叉树左右子数据框数据 B_L 和 B_R

1. 初始化 $\tilde{A}'$ 为空

2. 遍历 $j=1...K$，将 A_j 中的元素 a_i^j 按照 $\Delta S_{a_i^j}$ 降序排列，保持 $\tilde{A}$ 中除 j 维以外的元素添加到 $\tilde{A}'$ 中，然后依次添加 a_i^j 到 $\tilde{A}'$，并不断计算 $\rho(\tilde{A}',R)$ 直到最大。

不断重复上述步骤 1、步骤 2，直到 Suspiciousness 收敛。

将 $B_l=\tilde{A}'$，$B_r=R-B_l$。

2.2 算法时间复杂度分析

算法 2 局部搜索单密集块算法的时间复杂度是 $o(T\times K\times(E+N\log N))$,其中 T 是迭代次数，K 是张量数据的维度，E 是数据中非零元素的数量，N 是任意维度中最大的长度。通常情况下 T 和 K 是设置成常数，所以算法 2 的时间复杂度和张量数据中非零元素的个数成线性关系。算法 1 二叉树搜索算法的时间复杂度是 $o(c)$，c 是运算算法 2 的次数，所以算法 1 的时间复杂度和算法 2 的时间复杂度成线性关系。所以整个算法时间复杂度是 $o(c\times T\times K\times(E+N\log N))$。

2.3 二叉树生长条件

定义 3 假设初始数据是稀疏矩阵（即 $\lambda<<1$），对原数据进行算法 2 计算，可以得到 Suspiciousness 值最大的子张量数据，定义为左子树，编号为 1，将左子树从原数据切除，剩下的子张量数据定义为右子树，编号为 0。根数据进行算法 2 计算，可以得到左子树，如果根数据的编号含有 1，那么将左子树在根数据所在维度上的所有值剪除剩下的部分为右子树，如果根数据中的编号不含 0，那么将左子树从根数据中剪除。对数据进行基于原背景数据疑心度计算分别为 $key_{根}$、key_L 和 key_R。

假设 $e=key_{根}-(key_L+key_R)$，根据式(1)将其展开得到式(2)。

$$\begin{aligned}e=&s_L(\ln\frac{s_L}{S}-1)+S\frac{n_L}{N}-s_L\ln\frac{n_L}{N}+s_R(\ln\frac{s_R}{S}-1)+S\frac{n_R}{N}\\&-s\ln\frac{n_R}{N}-s_{根}(\ln\frac{s_{根}}{S}-1)-S\frac{n_{根}}{N}+s_{根}\ln\frac{n_{根}}{N}\end{aligned} \tag{2}$$

定理 1 如果根数据是完全异常或者完全非异常，那么 $key_{根}\ge key_L+key_R$。

证明

情况 1 假设根数据是完全异常，则 $\lambda_{根}\approx\lambda_L\approx\lambda_R>>\lambda$，而且是采用切割的方法，则 $n_L+n_R<n_{根}$。因为 $\lambda=\frac{s}{n}$，所以将式(2)化简得到式(3)。

$$e=(\lambda_{根}\ln\lambda_{根}-\lambda_{根}\ln\lambda-\lambda_{根}+\lambda)(n_L+n_R-n_{根}) \tag{3}$$

因为 $n_L+n_R-n_{根}<0$，

所以要证 $e<0$，只要证 $\lambda_{根}\ln\lambda_{根}-\lambda_{根}\ln\lambda-\lambda_{根}+\lambda>0$

将式(3)整理得到式(4):

$$e=\left(\ln\left(\frac{\lambda_{根}}{\lambda}\right)^{\lambda}+\ln\left(\frac{\lambda_{根}}{\lambda\cdot e}\right)^{\lambda_{根}-\lambda}\right)(n_L+n_R-n_{根}) \tag{4}$$

由于 $\lambda_{根}>>\lambda$，则 $\frac{\lambda_{根}}{\lambda}>1$，$\frac{\lambda_{根}}{\lambda\cdot e}>1$，

所以 $e>0$。

情况 2 如果根数据是完全非异常，

则 $\lambda_{根}\approx\lambda_L\approx\lambda_R\approx\lambda$，所以 $e\approx0$

综合情况 1 和 2 可以得到

$e=key_{根}-(key_L+key_R)\ge0$,即 $key_{根}\ge key_L+key_R$

证毕。

定理 2 如果根数据含有异常集合，则 $key_{根}<key_L+key_R$。

证明：将式(2)化简可得式 (5):

$$\begin{aligned}e=&s_L\ln\lambda_L+s_R\ln\lambda_R-s_{根}\ln\lambda_{根}-\ln\lambda(s_L+s_R-s_{根})\\&+(s_L+s_R-s_{根})-\lambda(n_L+n_R-n_{根})\end{aligned} \tag{5}$$

原数据为稀疏张量，所以有

$$s_L+s_R-s_{根}\approx\lambda(n_L+n_R-n_{根})\approx0。$$

对于包含异常集合的非完全异常块，必然存在非异常部分，因此有 $\lambda_L>\lambda_{根}$，且 $\lambda_R>\lambda_{根}$，所以

$$e\approx s_L\ln\lambda_L+s_R\ln\lambda_R-(s_L+s_R)\ln\lambda_{根}=s_L\ln\frac{\lambda_L}{\lambda_{根}}+s_R\ln\frac{\lambda_R}{\lambda_{根}}>0$$

证毕。

3 仿真实验

为了验证本文提出的算法的有效性，进行了大量实验，实验中首先对比了本文算法和相关异常集合检测的运行效率问题，然后重点对比本文提出的 DDB-DST 算法和文献[12]提出的 M-zoom 算法对于多异常集合检测的结果召回率和准确率分析。本文所有的算法均采用 R 语言编程实现，实验环境为 PC 机，装有 Intel(R)Core(TM)i5-2450M，CPU 频率 2.5 GHz 内存 4 GB，运行 Windows7 32 位操作系统。本文所采用的数据集在表 2 中给出，其中 LBNL 数据集来自 Lawrence Berkeley National Lab 提供的公开网络包，数据集 AirForce 数据集来自 1999KDDCup 的网络包数据。S-Data-2 和 S-Data-3 分别是含有若干异常集合的合成数据集。

3.1 合成数据集仿真

本节中通过合成数据集 S-Data-2D 和 S-Data-3D 来比较本

文算法和 M-zoom 算法对异常集合的检测效果。S-Data-2D：基于 ERP 模型根据如下参数生成随机数据集：(1)维度 K =2，(2)尺寸 500*500，(3)Possion 分布 λ=0.01。在随机生成的数据集中注入 3 个维度为 2 的异常集合，异常集合的尺寸分别为 10*10,20*20,30*30。任务是将这三个异常集合从随机生成的数据集中检测出来。表 3 给出了 CSBST 算法和 M-zoom 的检测结果。S-Data-3D:继续根据下列参数生成高维数据集：(1)维度 K=3，（2）尺寸 500*500*500，，(3) Possion 分布的 λ=0.01。在数据集中生成 3 个维度为 3 的异常集合，异常集合的尺寸分别为 15*10*15,10*10*10,20*10*30。表 3 给出了 DDB-DST 算法和 M-zoom 的检测结果。

表 2　数据集汇总

数据集	维度	大小
S-Data-2D	2	83.5K
S-Data-3D	3	512.3K
LBNL 数据集	4	3.73M
AirForce（10%）	7	64.6M

表 3　三种算法在合成数据集上的比较

数据集	S-Data-2D			S-Data-3D		
性能指标	准确率	召回率	F1	准确率	召回率	F1
DDB-BST	0.924	0.964	0.944	0.935	0.952	0.943
M-zoom	0.538	0.988	0.697	0.512	0.981	0.673
CrossSpot	0.560	0.540	0.550	0.610	0.524	0.563

表 4　用 DDB-BST 算法检测网络入侵结果

数据集	数据体积	连接次数	网络异常连接次数	网络异常所占比例
LBNL	3610*472*49*35	111253	110532	99.4%
	3610*81*1100*331	56848	55192	97.1%
AirForce	3×66×11×3×1×224×36	376229	375411	99.7%
	3×66×11×12×7×184×106	36791	35826	97.3%
	3×66×11×297×278×24×25	10241	9815	95.8%

表 5　DDB-BST 算法在真实数据集下同 M-zoom 和 CrossSpot 的比较

数据集	LBNL			AirForce		
性能评价	准确率	召回率	F1 值	准确率	召回率	F1 值
DDB-BST	0.986	0.973	0.974	0.940	0.982	0.961
M-zoom	0.612	0.998	0.759	0.524	0.987	0.685
CrossSpot	0.542	0.564	0.553	0.572	0.612	0.591

3.2　真实数据集仿真

通过两个真实的数据集证明了 DDB-BST 算法的有效性。LBNL 数据是由 4 个标称属性和 1 个数值属性组成。标称属性是时间(s)，源 IP 地址，目标 IP 地址和端口号。数值属性数据是发包数量。AirForce 数据集不同于 LBNL 数据集它并不包含 IP 信息，包含了七个属性分别是：protocol、service、src bytes、dst bytes、flag、court、srv count，数值属性是连接次数。

表 4 展示了在真实数据集中 DDB-BST 算法对于网络攻击检测的结果，可以发现异常集合通常是由多种网络攻击组成的，而在标称属性组成的高维张量数据中，发包数量或者是连接次数组成的密集部分的正是异常部分。表 5 展示了在 LBNL 和 AirForce 数据集中，DDB-BST 算法对其检测性能评价以及同 M-zoom 和 CrossSpot 的比较，从表中可以看出 DDB-DST 算法比 M-zoom 算法 F1 值提高了 20%，比 CrossSpot 算法 F1 值提高了 40%，主要是因为上述所示的两种方法只是找到评价指标最高的张量数据集，但评价指标最高并不能保证数据集中全是异常数据，而 DDB-BST 算法还要对这样的数据集进一步的判别来保证检测出的数据集中不含有异常数据。

4　结束语

本文研究了高维张量数据中多密集块的检测方法，提出了一种基于二叉树的多密集块检测算法，给出了二叉树生长的条件，并证明了条件的合理性。实验结果表明上述算法具有很高的准确率及召回率。

参考文献：

[1] Maruhashi K, Guo F, Faloutsos C. Multi aspect forensics: pattern mining on large-scale heterogeneous networks with tensor analysis [C]// Proc of International Conference on Advances in Social Networks Analysis and Mining. 2011: 203-210.

[2] Shah N, Beutel A, Gallagher B, et al. Spotting suspicious link behavior with fBox: an adversarial perspective [C]// Proc of Robot World Cup XVIII. Springer Interantional Publishing, 2014: 295-305.

[3] Rodrigues T, Cunha T, IencomD, et al. Retweet patterns: detection of spatio-temporal patterns of retweets [M]// New

[4] Advances in Information Systems and Technologies. [S. l.] : Springer International Publishing, 2016.

[5] 王越，张剑金，刘芳芳. 一种多特征微博僵尸粉检测方法与实 [J]. 中国科技论文, 2014 (1): 81-86.

[6] Shin K, Kang U. Distributed methods for high-dimensional and large-scale tensor factorization [C]// Proc of IEEE International Conference on Data Mining. 2015: 989-994.

[7] Papalexakis E E, Faloutsos C, Sidiropoulos N D. ParCube: sparse parallelizable tensor decompositions [J]. Acm Transc tions on Knowledge Discovery from Data, 2012, 10 (1): 521-536.

[8] Jeon I, Papalexakis E E, Kang U, et al. HaTen2: billion-scale tensor decompositions [C]// Proc of International Conference on Data Engineering. 2015: 1047-1058.

[9] Lee V E, Ning R, Jin R, et al. A survey of algorithms for dense subgraph

discovery [M]// Managing and Mining Graph Data 2010: 303-336

[10] Balalau O D, Bonchi F, Chan T H H, et al. Finding subgraphs with maximum total density and limited overlap [C]// Proc of the 8th ACM International Conference on Web Search and Data Mining. New York: ACM Press, 2015: 379-388.

[11] Sariyuce A E, Seshadhri C, Pinar A, et al. Finding the hierarchy of dense subgraphs using nucleus decompositions [C]// Proc of International Conference on World Wide Web. 2015: 927-937.

[12] 朱鎔，邹兆年，李建中. 不确定图上的Top-k稠密子图挖掘算 [J]. 计算机学报, 2016, 39 (8): 1570-1582.

[13] Bahmani B, Kumar R, Vassilvitskii S. Densest subgraph in streaming and MapReduce [J]. Computer Science, 2012: 454-465.

[14] Bahmani B, Goel A, Munagala K. Efficient Primal-dual graph algorithms for MapReduce [M]// Algorithms and Models for the Web Graph. Springer International Publishing, 2014: 59-78.

[15] Jiang M, Beutel A, Cui P, et al. A General Suspiciousness Metric for Dense Blocks in Multimodal Data [C]// IEEE International Conference on Data Mining. 2015: 781-786.

[16] Shin K, Hooi B, Faloutsos C. M-zoom: fast dense-block detection in tensors with quality guarantees [M]// Machine Learning and Knowledge Discovery in Databases. 2016

[17] Jiang M, Beutel A, Cui P, et al. Spotting suspicious behaviors in multimodal data: a general metric andalgortihms [J]. IEEE Trans on Knowledge & Data Engineering, 2016: 1